Effect of Router Buffers on Stability of Internet Congestion Control Algorithms

Somayeh Sojoudi
Steven Low
John Doyle

Oct 27, 2011
Resource allocation problem

Objective

Fair assignment of rates to the users to maximize the utility (Kelly, 1997):

\[
\begin{align*}
\max_{x_s \geq 0} \sum_{s \in S} U_s(x_s) \\
\sum_{s \in S(l)} x_s & \leq c_l, \quad \forall l \in \mathcal{L}
\end{align*}
\]
Congestion control: A decentralized solution to the resource allocation problem

Congestion control algorithms:

- Primal
- Dual
- Primal-dual
Primal-dual:

\[
\dot{x}_s(t) = k_s(x_s(t))(U'_s(x_s(t)) - q_s(t)), \quad \forall s \in S \tag{1}
\]

\[
\dot{p}_l(t) = h_l(p_l(t))(y_l(t) - c_l)_{p_l(t)}^+, \quad \forall l \in \mathcal{L}
\]

Dual:

\[
x_s(t) = U'_s^{-1}(q_s(t)), \quad \forall s \in S \tag{2}
\]

\[
\dot{p}_l(t) = h_l(p_l(t))(y_l(t) - c_l)_{p_l(t)}^+, \quad \forall l \in \mathcal{L}
\]
Communication from links to sources & vice versa:

- Queueing delay (implicit)

Communication from sources to links:

- Queue size (implicit)
Queue dynamics:

\[ \dot{b}_l(t) = (y_l(t) - c_l)_{b_l(t)}^+ \]

Link price dynamics:

\[ \dot{p}_l(t) = h_l(p_l(t))(y_l(t) - c_l)_{p_l(t)}^+, \quad \forall l \in \mathcal{L} \]

i. \( h_l(p_l(t)) = \frac{1}{c_l} \):
   \( p_l(t) \): Queueing delay

ii. \( h_l(p_l(t)) = K, k > 0 \):
    \( p_l(t) = kb_l(t) \) \( (b_l(t): \text{queue size of link } l) \).
Pricing mechanisms:

i. Queueing delay
ii. Queue size

Congestion control algorithms are globally asymptotically stable!
Pricing mechanisms:

i. Queueing delay
ii. Queue size

Congestion control algorithms are globally asymptotically stable!

Unrealistic assumption: input rate at every link = source rate
Pricing mechanisms:

i. Queueing delay
ii. Queue size

Congestion control algorithms are globally asymptotically stable!

Unrealistic assumption: input rate at every link = source rate

True model: input rate at link = output rate at previous link
Goal:

- Derive a more accurate model, which takes into account the effect of buffering on output flows.
- Study the stability of the dual and primal-dual algorithms.
New model of queue dynamics

- \( b_{ls}(t) \): Backlog at link \( l \) from source \( s \) at time \( t \).
- \( b_l(t) = \sum_{s \in S(l)} b_{ls}(t) \): Aggregate backlog at link \( l \) at time \( t \).

\[
\begin{align*}
\dot{b}_{l1}(t) &= x_{l1} - y_{l1} \\
\dot{b}_{l2}(t) &= x_{l2} - y_{l2} \\
\dot{b}_l(t) &= x_{l1} + x_{l2} - y_{l1} - y_{l2} \\
&= x_{l1} + x_{l2} - c_l
\end{align*}
\]
Define:

\[ \theta_{ls}(t) = \frac{\dot{b}_{ls}(t)}{\dot{b}_l(t)}, \quad l \in \mathcal{L}, \ s \in \mathcal{S}, \ t \geq 0 \]

\[ \theta_{ls}(t) \] for different service disciplines:

- Generalized Weighted Fair Queuing (WFQ)
- First in First out (FIFO)
Generalized WFQ

Source 1
\[ x_{11}(t) \quad w_1(t) \]

Source 2
\[ x_{12}(t) \quad w_2(t) \]

Link \( l \)
\[ y_{l1}(t) = \frac{w_1(t)}{w_1(t) + w_2(t)} C_l \]
\[ y_{l2}(t) = \frac{w_2(t)}{w_1(t) + w_2(t)} C_l \]

If \( w_i(t) = x_{li}(t) \), then

\[ \theta_{li}(x(t)) = \frac{x_{li}(t)}{\sum_{j=1}^{N} x_{lj}(t)}, \quad \forall i \in \{1, \cdots, N\} \]
First in First out:

\[ \theta_{ls}(t) = \frac{\dot{b}_{ls}(t)}{\dot{b}_l(t)} = \frac{x_{ls}(t) - \frac{\sum_{s' \in S(l)} x_{ls}(t - \tau(t))}{\sum_{s' \in S(l)} x_{ls'}(t - \tau(t))} \cdot c_I}{\sum_{s' \in S(l)} x_{ls'}(t) - c_I} \]
$\Theta(t) = \begin{bmatrix}
\cdots & \sum \theta_{is}(t) & \cdots \\
\cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots 
\end{bmatrix}_{L \times L}$

- Diagonal entries are equal to 1
- Nonsingular matrix by assumption
Queue dynamics

Theorem

The buffer occupancies satisfy the differential equation

\[ \dot{b}(t) = \left( (I - \Theta(t))b(t) + Rx(t) - c \right)_{b(t)}^+, \quad \forall t \geq 0 \]

If the vector \( b(t) \) is strictly positive, then

\[ \dot{b}(t) = \Theta(t)^{-1}(Rx(t) - c). \]
Stability of primal-dual and dual

Dual:

$$x_s(t) = U_s^{-1}(q_s(t)) \quad \forall s \in S$$

$$p_l(t) = h_l b_l(t) \quad \forall l \in \mathcal{L},$$

Primal-dual:

$$\dot{x}_s(t) = k_s(U'_s(x_s(t)) - q_s(t)) \quad \forall s \in S$$

$$\dot{p}_l(t) = h_l b_l(t) \quad \forall l \in \mathcal{L},$$

Queue dynamics

Common model:

$$\dot{b}_l(t) = (y_l(t) - c_l)_{b_l(t)}^+$$

More accurate model:

$$\dot{b}_l(t) = (\bar{y}_l(t) - c_l)_{b_l(t)}^+$$
Stability of primal-dual
Stability of primal-dual

- Linearization around the equilibrium point:

\[
\begin{bmatrix}
\dot{x}(t) \\
\dot{p}(t)
\end{bmatrix}
= \begin{bmatrix}
-Diag \left\{ \frac{w_1}{x_1^2}, \ldots, \frac{w_7}{x_7^2} \right\} & -R^T \\
\Theta^{-1} R & 0
\end{bmatrix}
\begin{bmatrix}
x(t) \\
p(t)
\end{bmatrix}
\]

- Unstable eigenvalues:

\[0.4060 \pm 1.7721i, \quad 0.2590 \pm 1.0134i, \quad 0.0279 \pm 0.6940i\]
Stability of primal-dual

- Simulation for initial values $\mathbf{b}(0) = \mathbf{p}^*$, $\mathbf{x}(0) = 1.001 \mathbf{x}^*$
Stability of primal-dual

**Theorem**

Given a constant $\Theta(t) = \Theta$, the following cases can occur for the primal-dual algorithm:

i) **All eigenvalues of $R^T\Theta^{-1}R$ are real and nonnegative:** The congestion algorithm is globally stable if $\Theta$ is symmetric.

ii) **At least one eigenvalue of $R^T\Theta^{-1}R$ is complex or negative real:** There exists a strictly positive number $\alpha$ such that the primal-dual algorithm becomes unstable if $U_s(x_s)$ is taken as $\alpha \log(x_s)$.

**Sketch of the proof:**

- Part i) Lyapunov function
  \[ V(x, p) = (x(t) - x^*)^T(x(t) - x^*) + (P(t) - P^*)^T\Phi(P(t) - P^*) \]

- Part ii) Decompose the linearization matrix
  \[ A = \begin{bmatrix} -\alpha D & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -R^T \\ \Phi^{-1}R & 0 \end{bmatrix} \]
  \[ \Rightarrow \text{det} \left( \lambda^2 I + R^T\Phi^{-1}R \right) = 0 \]
Stability of primal-dual

**Theorem**

Let $\mathbf{x}^*$ denote the solution of the resource allocation problem associated with the utility functions $U_1(x_1), \ldots, U_S(x_S)$. For a state-dependent $\Theta(t) = \Theta(\mathbf{x}(t))$, we have:

i) **All eigenvalues of** $R^T \Theta(\mathbf{x}^*)^{-1} R$ **are real and nonnegative:** The congestion algorithm is locally stable if $\Theta(\mathbf{x}^*)$ is symmetric.

ii) **At least one eigenvalue of** $R^T \Theta(\mathbf{x}^*)^{-1} R$ **is complex or negative real:** There exists a strictly positive number $\alpha$ such that the primal-dual algorithm becomes unstable if each user $s \in S$ takes its utility function as $\alpha U_s(x_s)$ rather than $U_s(x_s)$. 
Stability of dual

- No unstable example. Why?
  i. Θ is highly structured.
  ii. Dual has less dynamics involved compared to primal-dual.

**Theorem**

Given Θ(t) = Θ(x(t)), assume that Θ(x(t)) is a continuous function at the point x(t) = x*. The dual algorithm is locally stable, provided the matrix Θ(x*) + Θ(x*)^T is positive definite (if Θ(t) is constant, the algorithm is globally asymptotically stable).
Stability of dual and primal-dual

Theorem
The dual algorithm and primal-dual algorithm are both globally asymptotically stable with the unique equilibrium point \((\mathbf{x}^*, \Theta^{-1}\mathbf{p}^*)\), provided the source price vector \(\mathbf{q}(t)\) is taken as \(R^T\Theta\mathbf{p}(t)\) as opposed to \(R^T\mathbf{p}(t)\).
Remarks:

i. Explicit feedback and stability or Implicit communication and getting close to instability?

ii. General ideas w.r.t control and queueing (TCP, an illustrative example).

iii. The issues will presumably be present in any flow-based control (may be of increasing interest as we try to design new architectures).
Summary

- Derived a new model which takes into account the effect of buffering on output flows.

- Studied the stability of the congestion control algorithms under the new model (these algorithms might no longer be stable!).

- Provided sufficient conditions for the stability of dual and primal-dual algorithms.

- Proposed a new pricing mechanism for the stability of these algorithms.
GEOGRAPHICAL LOAD BALANCING

Zhenhua Liu, Minghong Lin
Adam Wierman, Steven Low
Caltech

Lachlan Andrew
Swinburne, Australia
Energy optimization

- Dynamic voltage scaling
- Spin down or park the disk
- Dynamic transmission power
- Application level power manager
- Dynamically turn on/off servers

Adam Wierman, Caltech
Geographical load balancing

Tradeoff between energy & delay

- Optimize over request routing and server capacity
- Balance
  - propagation delay (which DC?)
  - queueing delay (how much server capacity at DC?)
  - energy cost
- Exploit fluctuations of price & traffic over time & space
Mean arrival rate at time $t$ is $L_j(t)$.

Data Center $i$

- Total # of servers is $M_i$
- # of active servers is $m_i(t)$

Sum of all rates into $i$ equals to the load:

$$\sum_i \lambda_{ij}(t) = L_j(t)$$
Mean arrival rate at time $t$ is $L_j(t)$.

The goal is to minimize energy cost and delay costs:

$$\min_{m_i, \lambda} \sum_{i \in N} p_i m_i + \beta \sum_{j \in J} \lambda_{ij} \left( \frac{1}{\mu_i - \lambda_i/m_i} + d_{ij} \right)$$

Subject to:

$$\sum_{i \in N} \lambda_{ij} = L_j(t), \quad \forall j \in J$$

$$\lambda_{ij} \geq 0, \quad \forall i \in N, \forall j \in J$$

$$0 \leq m_i \leq M_i, \quad \forall i \in N$$
**Goal**

\[
\begin{align*}
\min_{m, \lambda} & \sum_{i \in N} p_i m_i + \beta \sum_{j \in J} \lambda_{ij} \left( \frac{1}{\mu_i - \lambda_i/m_i} + d_{ij} \right) \\
\text{s.t.} & \sum_{i \in N} \lambda_{ij} = L_j, & \forall j \in J \\
& \lambda_{ij} \geq 0, & \forall i \in N, \forall j \in J \\
& 0 \leq m_i \leq M_i, & \forall i \in N 
\end{align*}
\]

**Structural implications of optimality condition**

- Sources only choose data centers with the same, lowest marginal cost
- All datacenter utilizations are equalized
- Simplest (sparse) routing structure

Liu, Lin, Wierman, L., Andrew, Sigmetrics 2011
Setup: Workload

→ one source per state
→ workload scaled by population with Internet accesses
→ shift by time zones
→ propagation delay proportional to distance

Workload: I/O activity @ Microsoft Cambridge from 8 servers over a 48-hour period, starting at midnight PDT on Monday August 4, 2008.
Setup

Hotmail traffic trace
Aug 4, 2008
48-hr duration

48-hr of traffic traces for Hotmail

20 data centers (Google-like)

Industrial electricity price of each state in May 2010

Compare optimal strategy and min-delay strategy relative to min-load strategy
Optimizing delivery

• Global load balancing traditionally optimizes for server load or distance, not energy

• Route jobs to data centers with
  – min load
  – min distance from client
Optimizing delivery

• Our proposal: jointly minimize delay and energy consumption
  – must optimally balance load, distance, electricity cost

• Control actions available
  – Routing: which data center to serve a job/request
  – Server: how many servers to activate (vs in sleep mode)

• Exploit fluctuations in traffic & electricity prices
  – Both fluctuate across time and space
  – Optimally match traffic, requirements, prices
Example: optimizing delivery

<table>
<thead>
<tr>
<th>Conclusion</th>
<th>Optimal strategy</th>
<th>Min-delay strategy</th>
<th>Min-load strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall performance</td>
<td>Best</td>
<td>Good</td>
<td>Poor</td>
</tr>
<tr>
<td>Electricity cost</td>
<td>Best</td>
<td>Poor</td>
<td>Poor</td>
</tr>
<tr>
<td>Delay</td>
<td>Good</td>
<td>Best</td>
<td>Poor</td>
</tr>
</tbody>
</table>

prevalent strategy
Optimal strategy is 25% to 50% better overall relative to min-load strategy.
Optimal strategy incurs 25% - 65% lower cost relative to min-load strategy.

optimal strategy incurs 25% - 65% lower electricity cost

min-delay strategy incurs same electricity cost

hour
Optimal strategy incurs 30% smaller delay relative to min-load strategy.